1. A method of in-line processing a data packet while routing the packet through a router in a system transmitting data packets between a source and a destination over a network including the router, the method comprising:

receiving the data packet;

pre-processing layer header data for the data packet as the data packet is received and prior to transferring any portion of the data packet to packet memory;

storing the data packet in the packet memory;

determining a routing through the router including a next hop index describing the next connection in the network;

retrieving the data packet from the packet memory;

constructing a new layer header for the data packet from the next hop index while the data packet is being retrieved from memory; and

coupling the new layer header to the data packet prior to transfer from the router.

2. The method of claim 1 where the pre-processing step includes screening header layer data associated with the data packet for errors and dropping a bad data packet prior to transferring any portion of the data packet to packet memory.

3. The method of claim 2 where the screening includes screening Layer 2 (L2) and Layer 3 (L3) headers for errors.

4. The method of claim 3 where the L2 header is examined to detect errors arising from unrecognized L2

header formats and unconfigured L2 connections.

5. The method of claim 3 where the L3 header is examined to detect data packets with checksum errors, packet length errors and L3 header errors.

6. The method of claim 1 where the step of storing the data packet in memory includes dividing the data packet into cells of a fixed size and storing the cells in a distributed memory.

7. The method of claim 6 where the step of retrieving the data packet from memory includes reconstructing the packet from cells stored in the memory.

8. The method of claim 1 where the pre-processing includes stripping L2 header data from a data packet prior to storage the packet memory.

9. The method of claim 8 where the pre-processing step includes identifying the beginning of the L3 header and examining the L3 header for errors prior to the storage of the data packet in the packet memory.

10. The method of claim 1 where the next hop index is a pointer to a sequence stored in a storage device within the router and the step of constructing a new layer header includes executing the sequence.

11. The method of claim 10 where the execution of the sequence includes retrieving a common template for constructing a common portion of an L2 header to be attached

to the data packet and a custom template for constructing a unique portion of the L2 header.

12. The method of claim 11 where the common and unique templates are executable code operable to construct and associated portion of an L2 header for the data packet.

13. The method of claim 1 where the step of receiving the data packet includes receiving a plurality of data packets for processing from a plurality of input ports representing a plurality of streams of data to be routed through the router.

14. The method of claim 1 where the step of pre-processing the data packet includes dividing the data packet into fixed length cells and parsing the L2 header associated with the first cell of the data packet prior to receipt of the entire data packet.

15. The method of claim 14 where the step of parsing the L2 header includes examining the L2 header for errors and identifying the start of a next layer header in the data packet.

16. The method of claim 15 including temporarily storing the cells for a data packet in a queue after L2 header parsing is completed and spraying consecutive cells in a data packet to a different bank in the packet memory.

17. The method of claim 16 including snooping while the cells are being written to the queue and parsing the L3 header including examining the L3 header for errors.

18. The method of claim 16 including dropping a data packet if errors are detected in the L2 header during L2 header parsing without storing a cell associated with the data packet in the queue.

19. A router for in-line processing a data packet while routing the packet in a system transmitting data packets between a source and a destination over a network including the router, the router comprising:

a packet memory for storing portions of the data packet;

an input port for receiving a data packet including a header processing engine for evaluating header layer information upon receipt and prior to the storage of the data packet in the packet memory;

a controller for determining packet routing through the router including a next hop index indicating the next connection in the network for each data packet to be routed through the router; and

an output port for transferring the data packet to the destination including an output formatter for constructing a layer header for the data packet to facilitate the transfer of the packet to the destination.

20. The router of claim 19 where the header processing engine is operable to screen header layer data associated with the data packet for errors and drop a bad data packet prior to transferring any portion of the data packet to packet memory.

21. The router of claim 20 where the header

processing engine screens Layer 2 (L2) and Layer 3 (L3) headers for errors.

22. The router of claim 21 where the header processing engine examines the L2 header to detect errors arising from unrecognized L2 header formats and unconfigured L2 connections.

23. The router of claim 21 where header processing engine examine the L3 header to detect data packets with checksum errors, packet length errors and L3 header errors.

24. The router of claim 19 includes a cell packetizer operable to divide the data packet into cells of a fixed size prior to transfer to packet memory.

25. The router of claim 24 where the packet memory is a distributed memory and the router includes a spray engine for distributing cells across the distributed memory.

26. The router of claim 19 where the header processing engine strips L2 header data from a data packet prior to storage the packet memory.

27. The router of claim 26 where the header processing engine identifies the beginning of an L3 header and examines the L3 header for errors prior to the storage of the data packet in the packet memory.

28. The router of claim 19 where the next hop index is a pointer to a sequence stored in a storage device within the router and the output formatter executes the sequence to

construct a new layer header.

29. The router of claim 28 where the storage device includes a common template for constructing a common portion of an L2 header to be attached to the data packet and a custom template for constructing a unique portion of the L2 header.

30. The router of claim 29 where the storage device is distributed and includes a local portion within the output port and a remote portion elsewhere in the routing device and where the common template is stored in local portion and the custom template is stored in the remote portion.

31. The router of claim 29 where the common and unique templates are executable code operable to construct and associated portion of an L2 header for the data packet.

32. The router of claim 19 further including a plurality of input ports and a plurality of output ports, each of the input ports and output ports configurable to receive and process a plurality of data packets representing a plurality of streams of data to be routed through the router.

33. The router of claim 19 further including a cell packetizer operable to divide the data packet into fixed length cells and a L2 parsing engine operable to examine the L2 header associated with the first cell of the data packet prior to receipt of the entire data packet.

34. The router of claim 33 where the L2 parsing engine is operable to examine the L2 header for errors and identify the start of a next layer header in the data packet.

35. The router of claim 34 further including a queue operable for temporarily storing the cells for a data packet after L2 header parsing is completed and a spray engine operable to spray consecutive cells in a data packet to a different bank in the packet memory.

36. The router of claim 35 further including a L3 parsing engine operable to snoop while the cells are being written to the queue and parse the L3 header including examining the L3 header for errors.

37. The router of claim 36 where the L2 parser engine is operable to drop a data packet if errors are detected in the L2 header during L2 header parsing without storing a cell associated with the data packet in the queue.

38. A method of routing a data packet through a router in a system transmitting data packets between a source and a destination over a network including the router, the method comprising:

receiving the data packet;

dividing the data packet into cells of a fixed size;

while dividing the data packet, pre-processing layer header data for the packet to locate the beginning of a next layer header;

storing the cells in a memory;

prior to storing a first cell associated with the packet in the memory, screening header layer data for errors and dropping a bad packet prior to transferring any cells into the memory;

determining a routing through the router including a next hop index describing the next connection in the network;

reconstructing the packet from cells stored in the memory; and

constructing new layer header data for the packet from the next hop index.